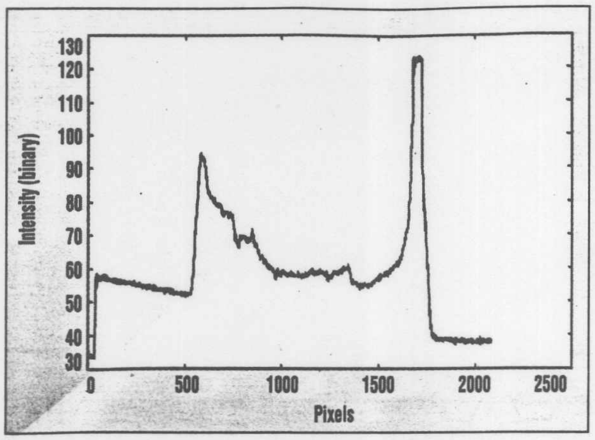


3. Here is a single video line obtained after scanning an image using the interface proposed in the article. Note that the first pixels are the sensor-reference "black" level.

For this approach, the software routines above must be moved to the IRQ7 (0x0f) interrupt handler. Alternately, an appropriate software interrupt, such as the 0x1C (timer interrupt), can be used instead of IRQ7. Finally, to use this interface as a camera, all that's required is the addition of a lens to the CCD sensor.

One line of the scanned image obtained using this architecture is shown in Figure 3. The code listing details an interrupt-driven "C" program that supports it. The performance limits of the proposed circuit are: pixel rate = 2 Mbytes/s; line rate = 1000 lines/s; signal-to-noise ratio = 41.25 dB.

Embed A Generator To Do Comm Test

Tom Seim
Systek, 415 N. Quay St., Kennewick, WA 99336; (509) 735-1200;
fax (509) 735-6534; e-mail: tom_systek@msn.com

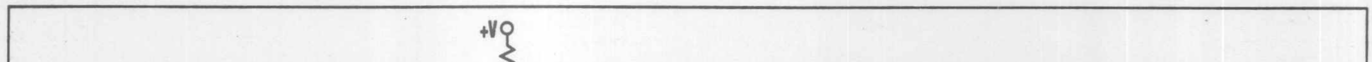
CIRCLE 521

When testing digital communication equipment, it's very useful to simulate the communication channel for various distortion effects, one of which is edge jitter. Instruments are available that will do this, but what if a simulator could be built right into the product for instant loopback testing? Presented here is a simple circuit that accomplishes this by using programmable logic commonly found in these products *(Fig. 1)*.

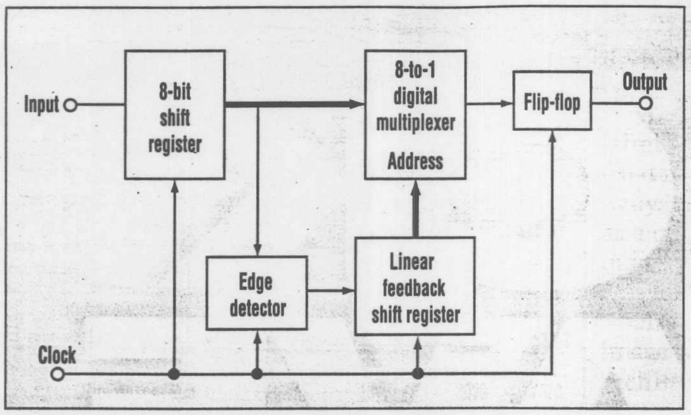
The circuit has four parts: a shift register, an edge detector, a multiplexer, and a linear feedback shift register (LFSR) *(Fig. 2)*. LFSRs are a special type of counter that uses very few components, primarily flip-flops and one or more exclusive-OR gates. The principal reason for using an LFSR in this circuit isn't its simplicity (although this is a nice side benefit), but rather for its pseudo-random count sequence. The jitter must be randomly distributed if it is to accurately simulate an actual communication channel; the LFSR count is used here to control a signal delay path.

The shift register and multiplexer are employed as a programmable digital signal delay. The channel address of the multiplexer selects the number of clock cycles that the input is delayed. The last part of the circuit is the edge detector, comprised of an exclusive-OR gate and flip-flop.

The last two output stages of the shift register feed the XOR gate. When a signal transition (on either a rising or a falling edge) reaches this stage of the shift register, the flip-flop will be



1. This jitter generator can be built right into the product for instant loopback testing, using less than 20% of a Xilinx XC3130A FPGA.



2. The block diagram outlines the arrangement of the four major parts of the jitter generator: a shift register, an edge detector, a multiplexer and a linear feedback shift register (LFSR).

set for just one clock cycle. This output is used as the clock enable for the LFSR, causing it to count once, and only once, for each signal edge.

The count occurs during an inactive portion of the input, ensuring that the delay won't change while the edge is "in" the shift register (this can cause unwanted glitches, which may themselves be useful but we're only interested in simulating jitter here). Finally, the output of the multiplexer is registered to eliminate switching glitches.

This circuit limits the jitter produced to eight discrete delay values. Additional delay values can be readily handled by increasing the size of the shift register and multiplexer.

The size of the LFSR is larger than the minimally necessary size in order to increase the randomness of the jitter sequence. A 3-bit LFSR repeats every seven clocks, which is readily apparent on a scope trace. This circuit uses a 7-bit LFSR with a count length of 127. However, it can be increased to as many stages as necessary for the required randomness.

A minor, but important, difference in this LFSR from conventional designs is the use of an exclusive-NOR gate (an exclusive-OR gate is used in standard designs). If the flip-flops are ever in the all-zero state, the counter will never exit that state (the XOR of 0 and 0 is 0). A simulator program was written to verify the count sequence and the even distribution of delay values.

The circuit was implemented in a Xilinx XC3130A using Orcad for schematic capture and the standard Xilinx Orcad library. The routed design used 13 logic cell blocks out of 100 available blocks. The only logic construct that may not translate directly into other FPGA and EPLD families is the clock enable. This can be worked around by either gating the clock input, or, if not possible, adding more feedback logic to each flip-flop:

$$D = D_{IN}\ \&\ CE + Q\ \&\ {\sim}CE$$

The scope traces show the output of a representative single trace *(Fig. 3a)* and several hundred traces accumulated in the envelope mode (the upper trace is the input and the lower trace is the output) *(Fig. 3b)*.

The amount, or percentage, of jitter is controlled by the clock frequency driving the circuit and the communications data rate.

$$Jitter\ (\%) = \frac{\text{Clock Period} * N * 100}{\text{Baud Period}}$$

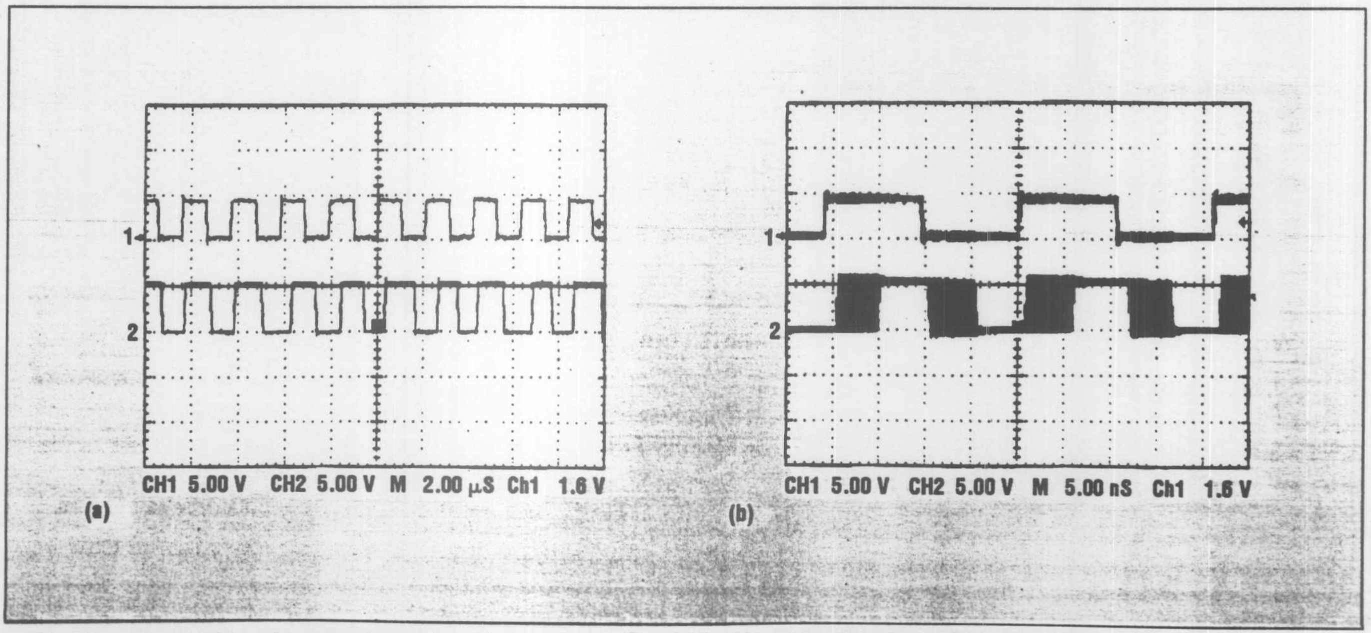
where N is the number of stages in the linear feedback shift register (eight in this example). Obviously, modifying the clock frequency can vary the jitter percentage.

ifd vote

Select your favorite Idea for Design in this issue and circle the appropriate number on the Reader Service Card. The winner receives a $300 Best-of-Issue award.

ideas wanted

Address your Ideas for Design submissions to: Ideas for Design Editor, Electronic Design, 611 Route 46 West, Hasbrouck Heights, NJ 07604 or e-mail to: mjames@penton.com.



3. The scope traces show the output of a representative single trace (a) and several hundred traces accumulated in the envelope mode (the upper trace is the input and the lower trace is the output with random signal jitter added at each input transition) (b).